Chapter 4
Configure SNMP

To configure SNMP, include the following statements at the [edit snmp] hierarchy level of the configuration.

```
snmp {
    access {
        context context-name {
            description description;
            group group-name {
                model usm;
                read-view view-name;
                security-level (none | authentication | privacy);
                write-view view-name;
            }
        }
        group group-name {
            model usm;
            user [ user-names ];
        }
        user [ user-names ] {
            authentication-password authentication-password;
            authentication-type (none | md5 | sha};
            privacy-password privacy-password;
            privacy-type (none | des);
            clients {
                address restrict;
            }
        }
    }
    community community-name {
        authorization authorization;
        clients {
            address restrict;
        }
        view view-name;
    }
    contact contact;
    description description;
    interface [ interface-name ];
    location location;
    name name;
    traceoptions {
        file size size files number;
        flag flag;
    }
    engine-id {
        local engine-id;
    }
    rmon {
        alarm index {
            description text-description;
            falling-event-index index;
            falling-threshold integer;
            interval seconds;
            rising-event-index index;
            rising-threshold integer;
            sample-type type;
            startup-alarm alarm;
            variable oid-variable;
        }
        event index {
            community community-name;
            description text-description;
            type type;
        }
    }
    traceoptions {
        file size size files number;
        flag flag;
    }
    trap-group group-name {
        categories category;
        destination-port <port-number>;
        targets {
            address;
        }
    version version;
    }

    trap-options {
        agent-address outgoing-interface;
        source-address address;
    }
    view view-name; {
        oid object-identifier (include | exclude)
    )
}
```

For information on configuring Remote Monitoring (RMON) alarms and events, see Chapter 11, “Configure RMON Alarm and Event Entries” on page 217.

For information on RMON alarm and event configuration statements, see Chapter 13, “Summary of RMON Alarm and Event Configuration Statements” on page 231.

By default, SNMP is disabled.

This chapter describes the minimum required configuration and discusses the following tasks for configuring SNMP:

Minimum SNMP Configuration

To configure the minimum requirements for SNMP, include statements at the [edit snmp] hierarchy level of the configuration:

```
[edit]
snmp {
    community public;
}
```

The community defined here as public grants read access to all MIB data to any client.

Configure the System Contact

You can specify an administrative contact for each system being managed by SNMP. This name is placed into the MIB II sysContact object. To configure a contact name, include the contact statement at the [edit snmp] hierarchy level:

```
[edit snmp]
contact contact;
```

If the name contains spaces, enclose it in quotation marks (" ").

Example: Configure the System Contact

Define the system contact:

```
[edit]
snmp {
    contact "Junipero Berry, (650) 555-1234";
}
```

Configure the System Location

You can specify the location of each system being managed by SNMP. This string is placed into the MIB II sysLocation object. To configure a system location, include the location statement at the [edit snmp] hierarchy level:

```
[edit snmp]
location location;
```

If the location contains spaces, enclose it in quotation marks (" ").

Example: Configure the System Location

Specify where the system is located:

```
[edit]
snmp {
    location "Row 11, Rack C";
}
```

Configure the System Description

You can specify a description for each system being managed by SNMP. This string is placed into the MIB II sysDescription object. To configure a description, include the description statement at the [edit snmp] hierarchy level:

```
[edit snmp]
description description;
```

If the description contains spaces, enclose it in quotation marks (" ").

Example: Configure the System Description

Specify the system description:

```
[edit]
snmp {
    description "M40 router with 8 FPCs";
}
```

Configure the System Name

Specify the system name override:

```
[edit snmp]
name name;
```

If the name contains spaces, enclose it in quotation marks (" ").

Example: Configure the System Name

Specify the system name override:

```
[edit]
snmp {
    name "snmp 1";
}
```

Configure the SNMP Community String

The SNMP community string defines the relationship between an SNMP server system and the client systems. This string acts like a password to control the clients' access to the server. To configure a community string, include the community statement at the [edit snmp] hierarchy level:

```
[edit snmp]
community name {
    authorization authorization;
    clients {
        default restrict;
        address restrict;
    }
    view view-name;
}
```

If the community name contains spaces, enclose it in quotation marks (" ").

The default authorization level for a community is read-only. To allow Set requests within a community, you need to define that community as authorization read-write. For Set requests, you also need to include the specific MIB objects that are accessible with read-write privileges using the view statement. The default view includes all supported MIB objects that are accessible with read-only privileges; no MIB objects are accessible with read-write privileges. For more information on the view statement, see "view" on page 211.

The clients statement lists the IP addresses of the clients (community members) that are allowed to use this community. If no clients statement is present, all clients are allowed. For the *address*, you must specify an address, not a hostname. Include the default restrict option to deny access to all SNMP clients for which access is not explicitly granted. We recommend that you always include the default restrict option to limit SNMP client access to the local router.

Examples: Configure the SNMP Community String

Grant read-only access to all clients. With the following configuration, the system responds to SNMP Get, GetNext, and GetBulk requests that contain the community string public:

```
[edit]
snmp {
    community public {
        authorization read-only;
    }
}
```

Grant all clients read-write access to ping MIB and jnxPingMIB. With the following configuration, the system responds to SNMP Get, GetNext, GetBulk, and Set requests that contain the community string private and specify an OID contained in the ping MIB or jnxPingMIB hierarchy:

```
[edit]
snmp {
    view ping-mib-view {
        oid .1.3.6.1.2.1.80 include;          # pingMIB
        oid .1.3.6.1.4.1.2636.3.7 include;  # jnxPingMIB
    community private {
        authorization read-write;
        view ping-mib-view;
        }
    }
}
```

The following configuration allows read-only access to clients with IP addresses in the range 1.2.3.4/24, and denies access to systems in the range of fe80::1:2:3:4/64:

```
[edit]
snmp {
    community field-service {
        authorization read-only;
        clients {
            default restrict;              # Restrict access to all SNMP clients not explicitly
                                           # listed on the following lines.
            1.2.3.4/24;                    # Allow access by all clients in 1.2.3.4/24; except
            fe80::1:2:3:4/64 restrict;# fe80::1:2:3:4/64
        }
    }
}
```

Overview of SNMP Trap Options

Some carriers have more than one trap receiver that forwards traps to one central NMS. This allows for more than one path for SNMP traps from a router to the central NMS through different trap receivers. A router can be configured to send the same copy of each SNMP trap to every trap receiver configured in the trap group. For more information on trap groups, see "trap-group" on page 209.

The source address in the IP header of each SNMP trap packet is set to the address of the outgoing interface by default. When a trap receiver forwards the packet to the central NMS, the source address is preserved. The central NMS, looking only at the source address of each SNMP trap packet, assumes that each SNMP trap came from a different source.

In reality, the SNMP traps came from the same router, but each left the router through a different outgoing interface.

The options discussed in this section are provided to allow the NMS to recognize the duplicate traps and to distinguish v1 SNMP traps based on outgoing interface.

Configure SNMP Trap Options

Using SNMP trap options, you can set the source address of every SNMP trap packet sent by the router to a single address regardless of the outgoing interface. In addition, you can set the agent address of the SNMPv1 traps. For more information on the contents of SNMPv1 traps, see RFC 1157.

To configure SNMP trap options, you can include the trap-options and trap-group statement at the [edit snmp] hierarchy level:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
    source-address address;
    }
trap-group group-name {
    categories category;
    destination-port <port-number>;
    targets {
        address;
    }
    version version;
}
```

Configure the Source Address for SNMP Traps

You can configure the source address of trap packets. Currently, the only value that can be specified is lo0. The value lo0 indicates the source address of the SNMP trap packets will be set to the lowest loopback address configured at the interface lo0.

To enable and configure the source address of SNMP traps, the source-address statement at the [edit snmp trap-options] hierarchy level must be included:

```
[edit snmp]
   trap-options {
      source-address address;
   }
```

To enable and configure the loopback address, the address statement at the [edit interfaces lo0 unit 0 family inet] hierarchy level must be included.

```
[edit interfaces]
lo0 {
   unit 0 {
      family inet {
         address ip-address;
      }
   }
}
```

Example: Configure the Loopback Address as the Source Address of Trap Packets

To configure the loopback address and source address trap option:

```
[edit snmp]
trap-options {
   source-address lo0;
   }
trap-group “urgent-dispatcher” {
   version v2;
   categories link startup;
   targets {
      192.168.10.22;
      172.17.1.2;
   }
}

[edit interfaces]
lo0 {
   unit 0 {
      family inet {
         address 10.0.0.1/32;
         address 127.0.0.1/32;
      }
   }
}
```

In this example, the IP address 10.0.0.1 is the source address of every trap sent from this router.

Configure the Agent Address for SNMP Traps

The agent address is only available in the SNMP v1 trap packets (see RFC 1157). By default, the router's default local address is used in the agent address field of the SNMPv1 trap. You can set the agent address by including the agent-address statement at the [edit snmp trap-options] hierarchy level. Currently the only option available is the address of the outgoing interface:

```
[edit snmp]
    trap-options {
        agent-address outgoing-interface;
    }
```

Example: Configure the Outgoing Interface as the Agent Address

To configure the outgoing interface as the agent address:

```
[edit snmp]
    trap-options {
        agent-address outgoing-interface;
    }
    trap-group “urgent-dispatcher” {
        version v1;
        categories link startup;
        targets {
            192.168.10.22;
            172.17.1.2;
        }
    }
```

In this example, each SNMP v1 trap packet sent will have its agent address value set to the IP address of the outgoing interface.

Configure SNMP Trap Groups

You can create and name a group of one or more types of SNMP traps and then define which systems receive the group of SNMP traps. The trap group must be configured for SNMP traps to be sent. To create an SNMP trap group, include the trap-group statement at the [edit snmp] hierarchy level:

```
[edit snmp]
trap-group group-name {
    categories category;
    destination-port <port-number>;
    targets {
        address;
    }
    version version;
}
```

The trap group name can be any string and is embedded in the community name field of the trap. To configure your own trap group port, use the destination port option. The default destination port is port 162.

Each trap group you define must have a name and one or more targets, which are the systems that receive the SNMP traps. Specify the targets by address, not by hostname, as follows:

- authentication—Authentication failures
- chassis—Chassis/environment notifications
- link—Link up-down transitions
- remote-operation—Remote operations
- rmon-alarm—Alarm for RMON events
- routing—Routing protocol notifications
- startup—System warm and cold starts
- vrrp-events—VRRP events such as new-master or authentication failures

Specify the types of traps the trap group can receive in the categories statement. For information on which traps belong to which category, see “Standard SNMP Traps” on page 169 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 185.

The version statement allows you to specify the SNMP version of the traps sent to targets of the trap-group. If you specify v1 only, SNMPv1 traps are sent. If you specify v2 only, SNMPv2 traps are sent. If all is specified, both an SNMPv1 and an SNMPv2 trap are sent for every trap condition. For more information on the version statement, see version on page 211.

Example: Configure SNMP Trap Groups

Set up a trap notification list named urgent-dispatcher for link and startup traps. This list is used to identify the network management hosts (1.2.3.4 and fe80::1:2:3:4) to which traps generated by the local router should be sent. The name specified for a trap group is used as the SNMP community string when the agent sends traps to the listed targets.

```
[edit]
snmp {
    trap-group "urgent-dispatcher" {
        version v2;
        categories link startup;
        targets {
            1.2.3.4;
            fe80::1:2:3:4;
        }
    }
}
```

Configure the Interfaces on Which SNMP Requests Can Be Accepted

By default, all router interfaces have SNMP access privileges. To limit the access through certain interfaces only, include the interface statement at the [edit snmp] hierarchy level:

```
[edit snmp]
interface [interface-names];
```

Specify the names of any logical or physical interfaces that should have SNMP access privileges. Any SNMP requests entering the router from interfaces not listed are discarded.

Example: Configure Secured Access List Checking

Grant SNMP access privileges only to devices on interfaces so-0/0/0 and at-1/0/1. The following example does this by configuring a list of logical interfaces:

```
[edit]
snmp {
    interface [ so-0/0/0.0 so-0/0/0.1 at-1/0/1.0 at-1/0/1.1 ];
}
```

The following example grants the same access by configuring a list of physical interfaces:

```
[edit]
snmp {
    interface [ so-0/0/0 at-1/0/1 ];
}
```

Configure MIB Views

By default, an SNMP community grants read access and denies write access to all supported MIB objects (even communities configured as authorization read-write). To restrict or grant read or write access to a set of MIB objects, you must configure a MIB view and associate the views with a community.

To configure MIB views, include the view statement at the [edit snmp] hierarchy level:

```
[edit snmp]
view view-name {
    oid object-identifier (include | exclude) ;
}
```

The view statement defines a MIB view and identifies a group of MIB objects. Each MIB object of a view has a common OID prefix. Each object identifier represents a subtree of the MIB object hierarchy. The subtree can be represented either by a sequence of dotted integers (such as .1.3.6.1.2.1.2) or by its subtree name (such as interfaces). A configuration statement uses a view to specify a group of MIB objects on which to define access. To enable a view, you must associate the view with a community.

To associate MIB views with a community, include the view statement at the [edit snmp *community-name*] hierarchy level:

```
[edit snmp community community-name]
view view-name;
```

Example: Ping Proxy MIB

Restrict the ping-mib community to read and write access of the ping MIB and jnxpingMIB only. Read or write access to any other MIB using this community is not allowed.

```
[edit snmp]
view ping-mib-view {
    oid .1.3.6.1.2.1.80 include;                #pingMIB
    oid jnxPingMIB include;                     #jnxPingMIB
}
community ping-mib {
    authorization read-write;
    view ping-mib-view;
}
```

For more information on the ping MIB, see RFC 2925 and "Juniper Networks Enterprise-Specific Extensions to the Ping MIB" on page 114.

Configure SNMPv3

To configure SNMPv3, you must perform the following tasks:

Configure the Local Engine ID on page 28

Configure SNMPv3 Access on page 29

Configure SNMP Views on page 30

Configure the Local Engine ID

The engine ID is the administratively unique identifier for the SNMP engine. To configure the local engine ID, include the engine-id statement at the [edit snmp] hierarchy level:

```
[edit snmp]
engine-id {
    local engine-id;
}
```

The local engine ID is defined as an SNMP v3 engine's administratively unique identifier and is used for identification, not for addressing. You must configure the local engine ID explicitly. The engine ID is in text format with its fifth octet equal to 4. If the engine ID is not configured, the system default IP address of the router is used as the default engine ID. The fifth octet of the default engine ID is 1.

Configure SNMPv3 Access

SNMPv3 access sets the SNMP access levels by context, group, and user. To configure the SNMPv3 access levels, include the context, group, and user configuration statements at the [edit snmp] hierarchy level:

```
[edit snmp]
access {
    context context-name {
        description description;
        group group-name {
            model usm;
            read-view view-name;
            security-level (none | authentication | privacy);
            write-view view-name;
        }
    }
    group group-name {
        model usm;
        user [ user-names ];
    }
    user [ user-names ] {
        authentication-password authentication-password;
        authentication-type (none | md5 | sha};
        privacy-password privacy-password;
        privacy-type (none | des);
        clients {
            address restrict;
        }
    }
}
```

Specify the *context-name* variable to set a collection of management information accessible by an SNMP entity. An SNMP entity can have access to many access contexts and therefore require text strings to identify each context. You must also associate a context with a specific access group and configure read and write views associated with each group. The access group configuration statement includes three security levels:

none—No security. SNMPv3 provides no authentication and no encryption functionality on any SNMP information.

authentication—Provides authentication capability but no encryption functions on any SNMP information.

privacy—Provides the authentication function and the encryption function on all SNMP information.

Specify the *group-name* variable to identify a collection of SNMP users that share the same access policy, in which object identifiers (OIDs) are read-accessible or write-accessible. Each group is the collection of users associated with the security model. You can only specify the model usm.

Specify the *user-name* variable to identify a person for whom management operations are performed and authorized. For each user, you can specify the authentication type, authentication password, privacy type, and privacy password. The values for authentication type are none, md5, and sha. The values for privacy type are none and des. SNMP implementations (and SNMP configuration applications) must ensure that passwords are at least 8 characters in length. All authentication and privacy passwords must be at least 8 characters.

Configure SNMP Views

SNMPv3 uses community-based views, the same as SNMPv1. For more information on how to configure community-based views for SNMPv3, see “Configure MIB Views” on page 27.

Example: Configure SNMPv3

```
snmp {
    view all {
    oid .1.3.6.1 include;
    }
    engine-id {
        local "isp-routers-0001";
    }
    access {
        user john {
            authentication-type md5;
            authentication-password "auth-secret-password";
            privacy-type  des;
            privacy-password "priv-secret-password";
        }
    group admin {
        user john
        model usm;
        }
    context router {
        description "router context";
        group admin {
            model usm;
            security-level privacy;
            read-view all;
            write-view all;
            }
        }
    }
}
```

Configure IPv6 over SNMP

You can configure IPv6 over SNMP for SNMP agents and SNMP traps. To configure IPv6 over SNMP for SNMP agents, include the community statement at the [edit snmp] hierarchy level:

```
[edit snmp]
community community-name {
    clients {
        address restrict;
    }
}
```

To configure IPv6 for SNMP for SNMP traps, include the trap-group configuration statement at the [edit snmp] hierarchy level:

```
[edit snmp]
trap-group group-name {
    targets
        address;
    }
}
```

Example: Configure IPv6 over SNMP

Configure SNMP agents:

```
[edit]
snmp {
    community cm1 {
        clients {
            1.2.3.4/24;
            f8019:abcd::1/64;
        }
    }
}
```

Configure SNMP traps:

```
[edit]
snmp {
    trap-group group1 {
        targets {
            1.2.3.4;
            f8019:abcd::1;
        }
    }
}
```

Trace SNMP Activity

To trace SNMP activity, include the traceoptions statement at the [edit snmp] hierarchy level:

```
[edit snmp]
traceoptions {
    file size size files number;
    flag flag;
}
```

The output of the tracing operations is placed into log files in the /var/log directory. Each of these log files is named after the SNMP agent that generates it. Currently, the following logs are created in the /var/log directory when the traceoptions statement is used:

chassisd

craftd

ilmid

mib2d

rmopd

serviced

snmpd

You can use the file statement to control log file generation. The size statement limits the size (in kilobytes) of each log file before it is closed and compressed and a new file opened in its place. The file statement limits the total number of log files archived for each SNMP agent.

You can specify one or more of the following values for the flag option:

all—Trace all SNMP events

general—Trace general events

interface-stats—Trace physical and logical interface statistics

pdu—Trace SNMP request and response packets

protocol-timeouts—Trace SNMP response timeouts

routing-socket—Trace routing socket calls

subagent—Trace subagent restarts

timer—Trace internal timer events

varbind-error—Trace variable binding errors

Example: Trace SNMP Activity

Trace information on SNMP packets:

```
[edit]
snmp {
    traceoptions {
        file size 10k files 5;
        flag pdu;
        flag protocol-timeouts;
        flag varbind-error;
    }
}
```